CS65: Introduction to Computer Science

Conditional Statements (if-elif-else)
Complex Boolean Expressions

Md Alimoor Reza
Assistant Professor of Computer Science

What to work on

Lab #2 due on Tuesday, September 23rd for sections#1 and#2. Wednesday, September 24th for sections#3

Text Reading

https://www.brianheinold.net/python/python_book.html#chapter_ifstatements

Review from last lecture

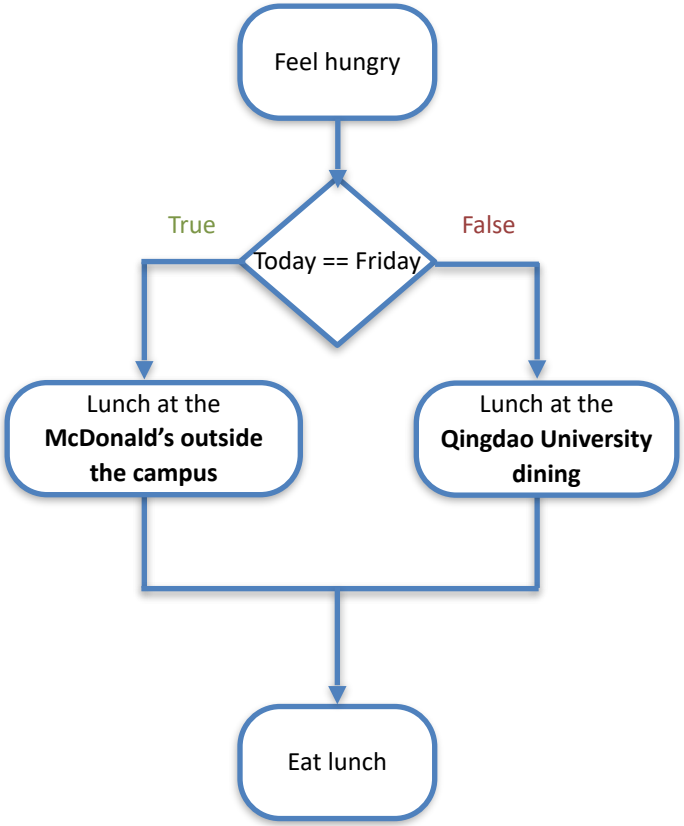
- if statement

- if-else statement

- Nested if-else statement

Review: Motivation

- program taking one *path* or *branch* of the code instead of taking another, based on the **boolean expression**'s value
- It allows us to ask true/false questions in our code. Depending on the boolean answer (True or False), the program will execute a specific branch.



Review: Syntax for if statements

- keyword **`if`**
- a condition, something that can be *true* or *false*
- colon **:**
- *indented* block of code
  - All of the code that is indented will be executed if the condition is true

if_statement.py

if_statement.py

```
1  # Alimoor Reza
2  # 09/18/2025
3  # simple if statement
4
5
6  num = int(input("Enter a number please: "))
7
8  if num == 5:
9      print("You won the lottery ...")
10
11 print("Thank you!")
```

Review from last lecture

- if statement
- if-else statement
- Nested if-else statement

Review: If statements and If-else statements

An if-statement will only execute the indented code if the condition is True

If the condition is False, the code block is *not* executed.

IF you want to have something different happen when the condition is False, then you can add an `else` clause.

Introducing,.... The If-Else statement

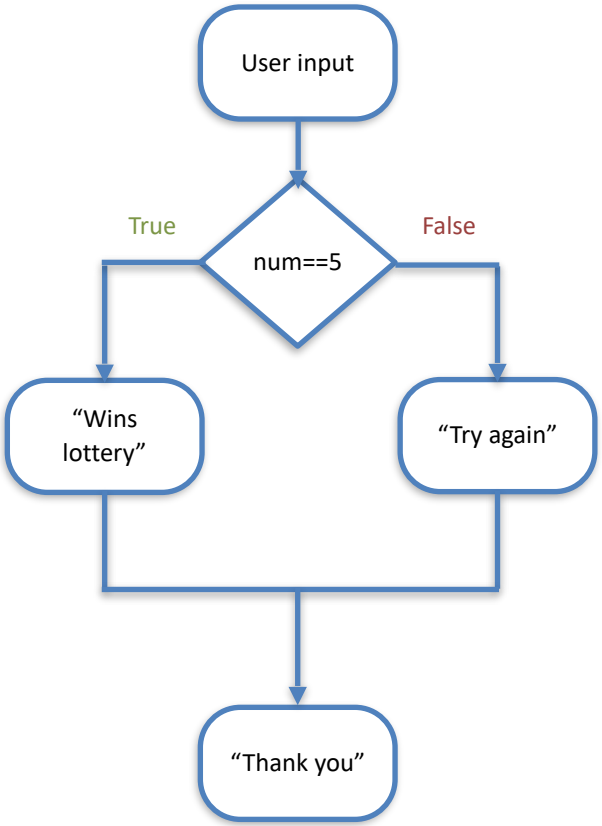
Review: `if` … `else` Statement

if_else_statement1.py

```
1  # Alimoor Reza
2  # 09/18/2025
3  # simple if-else statement
4
5
6  num = int(input("Please, enter a number. "))
7
8  if num == 5:
9      print("Yeah! I won a lottery ...")
10 else:
11     print("Oh gosh! better luck next time ...")
12
13 print("Thank you!")
14
```

Shell

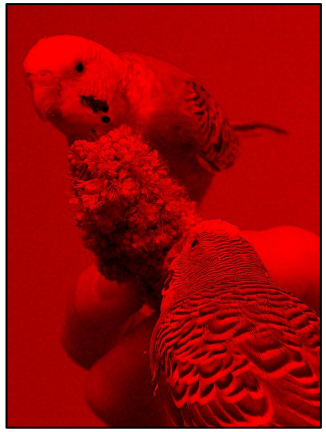
```
>>> %Run if_else_statement1.py

  Please, enter a number. 6
  Oh gosh! better luck next time ...
  Thank you!

>>>
```



Review: Syntax for if-else statements

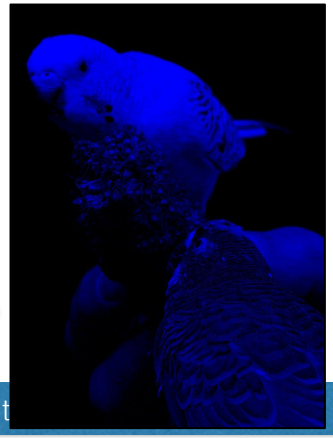
- keyword **`if`**
- a condition, something that can be *true* or *false*
- colon `:`
- *indented* block of code
  - All of the code that is indented here will be executed if the condition is **True**
- keyword **`else`**
- colon `:`
- indented block of code
  - All of the code that is indented here will be executed if the condition is **False**

Exercise #1



Red

Green



Blue

Exercise #1

Prompt the user to enter a string.

when user enters “Red” then print(“I like red channel of an RGB image.”)
when user enters “Green” then print(“I like green channel of an RGB image.”)
when user enters “Blue” then print(“I like blue channel of an RGB image.”)
when user enters anything else then print(“Image format is not correct.”)

Use ONLY if-elif-else statements to print out an appropriate response based on the inputted number and the above conditions

Review from last lecture

- if statement
- if-else statement
- Nested if-else statement

Review: Example

Imagine a situation where, given an input, you need to return a grade:

```
Prompt user for their score (between 0 and 100)
If score >= 90, print “you earned an A”
If score >= 80, print “you earned a B”
If score >= 70, print “you earned a C”
If score >= 60, print “you earned a D”
If score < 60, print “you earned an F”
```

Review: Consider this solution. What happens when I enter 90?

grader_if_else_wrong_solution.py

grader_if_else_wrong_solution.py | grader_nested_if_else_solution.py

```
1  # Alimoor Reza
2  # 09/23/2025
3  # multiple if else statements for grader
4  # this is incorrect solution
5
6  score = int(input("Enter your score: "))
7
8  if score >= 90:
9      print("Your grade is A")
10 if score >= 80:
11     print("Your grade is B")
12 if score >= 70:
13     print("Your grade is C")
14 if score >= 60:
15     print("Your grade is D")
16 else:
17     print("Your grade is F")
18
```

Shell

```
>>> %Run grader_wrong.py

  Enter your score: 90
Your grade is A
Your grade is B
Your grade is C
Your grade is D
```

Review: Is this better? Using nested if-else statements?

grader_nested_if_else_solution.py

```
2
3  score = int(input('Enter your score:'))
4
5  if score >= 90:
6      print("Your grade is A")
7  else:
8      if score >= 80:
9          print("Your grade is B")
10     else:
11         if score >= 70:
12             print("Your grade is C")
13         else:
14             if score >= 60:
15                 print("Your grade is D")
16             else:
17                 print("Your grade is F")
18
```

Today's content

- if-elif-else statement

- Complex Boolean expression

The if-elif-else Statement

if-elif-else statement: special version of a decision structure

Makes logic of nested decision structures simpler to write

Can include multiple `elif` statements

Syntax:

```
if condition1:
  statements
elif condition2:
  statements
else:
  statements
```

The if-elif-else Statement

```
1  # Alimoor Reza
2  # 09/23/2025
3  # multiple if-elif-else statements for grader
4  # this is correct solution
5  # this solution is better than nested if
6
7  score = int(input("Enter your score: "))
8
9  if score >= 90:
10     print("Your grade is A")
11 elif score >= 80:
12     print("Your grade is B")
13 elif score >= 70:
14     print("Your grade is C")
15 elif score >= 60:
16     print("Your grade is D")
17 else:
18     print("Your grade is F")
```

Exercise #2

Prompt the user for the number of hours of sleep they got last night.

Use ONLY if-elif-else statements to print out an appropriate response based on the inputted number and the following chart

Above 8	“You are well-rested!”
Between 4 and 8	“The coffee shop is around the corner.”
Between 0 and 4	“Are you sure you are awake?”
Less than 0	“input error.”

Today's content

- if-elif-else statement

- Complex boolean expression

Review: `bool` Data Type

- Notion of something being *true* or something being *false* can be represented with two `bool` types in Python
  - `True`
  - `False`
- In real life, we always encounter question with `Yes` or `No` answer
- Allows us to evaluate *true* or *false* question

Condition and Boolean expression

The condition for an if/if-else statement can be anything that results in a Boolean.

An expression, like `height < 48` that results in a Boolean is called a *Boolean expression.*

Some Boolean operators

- < less than
- > greater than
- <= less than or equal
  - note there is no ≤ button on your keyboard
- >= greater than or equal
- == equal (comparison)
  - note that = is used for assignment, they're different operators, don't confuse them!
- != not equal

Comparison Operators

- We can write expression that can be evaluated to a boolean value with other comparison operators
  - Compare two values or check something

Description	Example	Result
Less than	2 < 15	True
Greater than	2 > 15	False
Less than or equal	2 <= 15	True
Greater than or equal	2 >=15	False
Equality check	2 == 15	False
Inequality check	2 != 15	True

Complex Boolean Expression

- Expressions that are evaluated to two **`bool`** types
- Operations with logical operators — `and/or/not`
  - `and` – given two boolean, are both True? answer is True
    boolean expression$_1$ `and` boolean expression$_2$
  - `or` – given two booleans, at least one is True? answer is True
    boolean expression$_1$ `or` boolean expression$_2$
  - `not` – given a boolean expression, switch between True/False
    `not` boolean expression

Logical Operators

x	y	x and y
False	False	False
False	True	False
True	False	False
True	True	True

- expression$_1$ and expression$_2$

x	y	x or y
False	False	False
False	True	True
True	False	True
True	True	True

expression$_1$ or expression$_2$

x	not x
False	True
True	False

not expression

Let's Do More Boolean Expressions

X	Y	X and Y
2 < 15	2 >=15	False
3 < 15	2 ==15	False
3 < 15	15 == 15	True
16 > 15	2 != 15	True

- $expression_1$ and $expression_2$

Example: Consider the problem

Prompt the user to enter an integer number. Then it determines the following:

5, 10, 15, etc	**Odd and divisible by 5**
7, 14, 21, etc	**Odd and divisible by 7**
19, 38, 57, etc	**Odd and divisible by 19**

Use ONLY if-elif-else statements to print out an appropriate response based on the inputted number and the above conditions

5, 10, 15, etc	**Odd and divisible by 5**
7, 14, 21, etc	**Odd and divisible by 7**
19, 38, 57, etc	**Odd and divisible by 19**

Example: Consider the problem

example_if_elif_else.py

```
1  # alimoor reza
2  # 09/23/2025
3
4  number = int(input("Enter an integer number: "))
5
6  if number%2 == 1 and number%5 == 0:
7      print(number, " is odd and divisible by 5")
8  elif number%2 == 1 and number%7 == 0:
9      print(number, " is odd and divisible by 7")
10 elif number%2 == 1 and number%19 == 0:
11     print(number, " is odd and divisible by 19")
12 else:
13     print(number, " doesn't satisfy our criteria")
```

Exercise #3

Prompt the user to enter the age of a person

Infant	From birth up to about 1 year old
Toddler	If the age is between 1 to 3 years old
Child	If the age is between 4 to 12 years old
Teenager	If the age is between 13 to 19 years old
Adult	If the age is between 20 to 39 years old
Middle-aged	If the age is between 40 to 59 years old
Senior	If the age is 60 and above years old

Use ONLY if-elif-else statements to print out an appropriate response based on the inputted number and the above conditions

Summary

- **Takeaway from this lecture**
  - Encountering question with **yes** or **no** answer can be expressed with Python **Boolean** datatype, which has **true** or **false** values.
    - Boolean expression with logical operator (and, or, not)
    - Boolean expression with comparison operator (<, <=, >, ==, etc)
  - Selection statements are useful for branching inside your program
    - if block
    - if-else block
    - if - elif - elif - … - else block
- **Announcements:**
  - Lab 3 will be released soon
  - **Next Tuesday (09/30), there will be our first Content Quiz on *variable*, *expression*, *boolean expression* and *selection statements*.**